The Automatic Component of the LINGSTAT Machine-Aided Translation System*

Jonathan Yamron, James Cant, Anne Demedts, Taiko Dietzel, and Yoshiko Ito

Dragon Systems, Inc., 320 Nevada Street, Newton, MA 02160

ABSTRACT

We present the newest implementation of the LINGSTAT machine-aided translation system. The most significant change from earlier versions is a new set of modules that produce a draft translation of the document for the user to refer to or modify. This paper describes these modules, with special emphasis on an automatically trained lexicalized grammar used in the parsing module. Some preliminary results from the January 1994 ARPA evaluation are reported.

1. INTRODUCTION

LINGSTAT is an interactive machine-aided translation system designed to increase the productivity of a translator. It is aimed both at experienced users whose goal is high quality translation, and inexperienced users with little knowledge of the source whose goal is simply to extract information from foreign language text. (For an introduction to the basic structure of LINGSTAT, see [1].)

The first problem to be studied is Japanese to English translation with an emphasis on text from the domain of mergers and acquisitions, although recent evaluations have included general newspaper text as well. Work is also progressing on a Spanish to English system. The approach described below represents the current state of the Japanese system, and will be applied with minimal changes to Spanish.

Due to the special difficulties presented by the Japanese writing system, previous versions of LINGSTAT have focused on developing tools for the lexical analysis of Japanese (such as tokenization of the Japanese character stream, morphological analysis, and katakana transliteration), and on providing the user access to lexical information (such as pronunciations, glosses, and definitions) via online lookup tools. In addition, a simple parser was incorporated to identify modifying phrases. No translation of the document was provided. Instead, the user used the results of the above analyses and the online tools to construct a translation.

In the newest version of LINGSTAT, the user is provided with a draft translation of the source document. For a source language similar to English, a starting point for such a draft might be a word-for-word translation, but because Japanese word order and sentence structure are so different from English, a more general framework has been constructed. The translation process in LINGSTAT consists of the following steps:

- Tokenization and morphological analysis
- Parsing
- Rearrangement of the source into English order
- Annotation and selection of glosses via an English language model

These modules are described in Section 2 below. Section 3 gives some preliminary results from the January 1994 evaluation, and Section 4 discusses some plans for future improvements.

2. IMPLEMENTATION

Tokenization/de-inflection

In LINGSTAT, "tokenization" refers to the process of breaking a source document into a sequence of root words tagged, if necessary, with inflection information. For most languages, the tokenizer is basically an engine that oversees the de-inflection of source words into root forms. For languages like Japanese, written without spaces, the tokenizer also has the job of segmenting the source.

To segment Japanese, the LINGSTAT tokenizer uses a probabilistic dynamic programming algorithm to break up the character stream into the sequence of words that maximizes the product of word unigram probabilities, as supplied from a list of 300,000 words. Inflected forms are recognized during tokenization by a de-inflector module. This module has a language-independent engine driven by a language-specific de-inflection table. (More details on the function of these components can be found in [1].)

There have been two improvements in the tokenizer/de-

*This work was sponsored by the Advanced Research Projects Agency under contract number J-FBI-91-239.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1994		**00-00-1994 to 00-00-1994**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Automatic Component of the LINGSTAT Machine-Aided Translation System	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Dragon Systems Inc,320 Nevada Street,Newton,MA,02160**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**6**	

inflector module in the newer versions of the system, made possible by the introduction of part of speech information into the word list. The first is an extra check on the validity of suggested de-inflections by demanding consistency between the inflection and the part of speech of the proposed root. This has cleanly eliminated a number of spurious de-inflections that were previously handled in a more ad-hoc fashion. The second improvement, motivated more by plans to move on to Spanish, is to stop the tokenizer from attempting to uniquely specify the de-inflection path (and now, part of speech) for each token it finds. As an example of the problem this addresses, consider the two de-inflections of the Spanish word *ayudas*:

ayudas → *ayuda* (help, aid)

ayudas → *ayudar* (to help, to aid)

The original tokenizer made a choice between the noun and verb de-inflection based on the unigram frequency of the root. The new tokenizer still finds all allowed possibilities, but now simply passes them to the parser, which is better equipped to resolve the ambiguity.

Parsing

The parser in LINGSTAT has two roles. In the interactive component, information about modifying phrases is extracted from the parse and presented to the user as an aid to understanding the structure of each Japanese sentence. In the automatic component, the parse is the basis for the rearrangement of the Japanese sentence into English.

Because it is a long-term goal to have a system that can be quickly adapted to new domains and languages, a high priority is placed on developing parsing techniques that are capable of extracting some information automatically through training on new sources of text, thus minimizing the amount of human effort. In the current system, this has led to a two-stage parsing process. The first stage implements a coarse probabilistic context-free grammar of a few hundred human-supplied rules acting on parts of speech. Because of this coarseness, some parsing ambiguities remain to be resolved by the second-stage parser, which implements a simple, lexicalized, probabilistic context-free grammar trained on word co-occurrences in unlabeled Japanese sentences without human input.

Context-free parser. The first-stage parse is done using a standard probabilistic context-free grammar acting on about 50 parts of speech. Any ambiguities in part of speech assignments or de-inflection paths passed by the tokenizer/de-inflector are resolved based on the probability of possible parses. The grammar is allowed to contain unitary and null productions, which impose an ordering on the summation over rules that takes place during training; because there are currently only a few hundred rules, this ordering is checked by hand. The grammar can be trained with either the Inside-Outside [2] or Viterbi [3] algorithm.

It is essential that the parser return a parse, even a bad one, for subsequent processing. Therefore special, low-probability "junk" rules have been introduced to handle unanticipated constructions. These junk rules affect the generation of terminal symbols and take the following form: for each rule in which a non-terminal generates a particular terminal, a rule is added permitting the same non-terminal to generate any other terminal with a small probability. This allows the grammar to force the terminal string into a sequence that has a recognizable parse, but at a high enough cost such that any parse without such coersion will be favored. One advantage of this approach is that the grammar can compensate for missing or mislabeled data. Consider the fragment

$$\text{the}_{det}\ \ \text{large}_{adv}\ \ \text{dog}_{noun}$$

in which the adjective *large* has been mislabeled as an adverb. The junk rule permits the grammar to change its part of speech to something more appropriate provided no other sensible parse can be found.

In principle, the probability of invoking the junk rule could be trained with the other rules in the grammar (the example above suggests that it might be advantageous to do so). Currently this is not being done, based on the observation that an invocation of the junk rule is more likely an indication of a deficiency in the grammar than a useful correction to the data.

Lexicalized parser. The grammar implemented by the context-free parser is not fine enough to properly resolve certain kinds of ambiguity, such as the correct attachment of prepositional phrases or noun modifiers. These attachment problems are handled by a second parser, which does a top-down rescoring of certain probabilities computed in the first stage. Currently this rescoring is used to fine-tune attachments of particle phrases in Japanese sentences.

The second parser makes use of a second probabilistic grammar, one whose basic elements are the words themselves, and whose data consist of the probabilities of each word in the vocabulary to be generated in the context of any other word. Like a bigram language model, these probabilities can be trained on word co-occurrences in unlabeled sentences, but unlike bigrams, the grammar can learn about associations between words in a sentence regardless of their separation.

This very simple context-free grammar can be described as follows. To each word in the vocabulary we associate a terminal symbol w (the word itself) and a non-terminal symbol A_w. The grammar consists of the following two kinds of rules:

$$A_{w_1} \to A_{w_2} w_2 A_{w_2} A_{w_1} \ , \tag{1a}$$

$$A_{w_1} \to \phi \ , \tag{1b}$$

where ϕ represents the null production. In addition, we introduce a sentence start symbol A_0 with the production

$$A_0 \to A_w w A_w \ . \tag{2}$$

The probability of invoking a particular rule depends only on the word associated with the generating non-terminal and the terminal word in the production. The probabilities for (1a) and (1b) can therefore be written $p(w_1 \to w_2)$ and $p(w_1 \to \phi)$, respectively, and these satisfy

$$p(w_1 \to \phi) + \sum_{w_2} p(w_1 \to w_2) = 1 \ .$$

For the start symbol, the probabilities are $p(0 \to w)$ and satisfy

$$\sum_{w} p(0 \to w) = 1 \ .$$

There is no null production for the start symbol.

Roughly speaking, this grammar generates a sentence in the following manner. The start symbol first generates some word in the sentence. This word then generates some number of words to its left and right, which in turn generate other words to their left and right. From the form of the grammar it can be deduced that these generations are "local," in the sense that if w_1 generates w_2 on its right, w_2 is not allowed to generate any word to the left of w_1 (and similarly for w_1 generating w_2 on its left). The process continues in a cascading fashion until the whole sentence has been generated. The fertility of a particular word w (*i.e.*, the number of words it will typically generate) is determined by the probability $p(w \to \phi)$, as can be seen from examining the productions (1): a non-terminal A_w will continue to produce words through rule (1a) via tail recursion until rule (1b) is invoked.

Although this grammar has the same type and number of parameters as a bigram model, here they have a very different interpretation: they measure the probability of one word to generate another anywhere in the sentence, subject only to the constraints imposed by the generation process described above. Thus an association between two words that might typically appear together, such as *fast* and *car*, will be recognized even if another word might occasionally intervene, such as *red*. Another feature is that words with the most predictive power in a sentence tend to generate words with less predictive power, which has the consequence that words like *the* tend to generate no words at all. This is an improvement over a bigram model in which *the* is required to select a succeeding word from a distribution that is essentially flat across a large portion of the vocabulary.

This grammar shares the appealing feature of n-gram models that its parameters can be trained on unlabeled text (consisting of whole sentences). In this case, however, the training procedure is iterative—a modification of the Inside-Outside algorithm that is of order N^4 in the sentence length.[1] The iteration starts from a flat distribution, with co-occurrences of words within sentences leading to enhanced probabilities for some words to generate others.

The N^4 algorithm actually applies to a slightly different (but generatively equivalent) grammar than the one defined by rules (1) and (2). To implement this algorithm, we first replace rule (1a) by

$$A_{w_1} \to A_{w_2} w_2 A_{w_2} A_{w_1} \qquad (w_2 \text{ to the left of } w_1) \ ,$$

$$A_{w_1} \to A_{w_1} A_{w_2} w_2 A_{w_2} \qquad (w_2 \text{ to the right of } w_1) \ ,$$

where the probability of both rules is the same and given by $p(w_1 \to w_2)$. The only difference between this and rule (1a) is that when A_w generates multiple words to the right of w, they are generated right to left instead of left to right.

As an example of how the N^4 dependence arises, consider the inside calculation for this model. For a sentence $w_1 \dots w_N$, the quantities of interest for the inside pass are the probabilities $I(A_{w_i} \to w_j \dots w_{i-1})$ for $j < i$ and $I(A_{w_i} \to w_{i+1} \dots w_j)$ for $j > i$. These may be calculated recursively by the following formulae:

$$\begin{aligned} I(A_{w_i} \to w_j \dots w_{i-1}) = & \sum_{k=j}^{i-1} \sum_{l=k}^{i-1} p(w_i \to w_k) \\ & \times I(A_{w_k} \to w_j \dots w_{k-1}) I(A_{w_k} \to w_{k+1} \dots w_l) \\ & \times I(A_{w_i} \to w_{l+1} \dots w_{i-1}) \ , \end{aligned} \tag{3a}$$

$$\begin{aligned} I(A_{w_i} \to w_{i+1} \dots w_j) = & \sum_{k=i+1}^{j} \sum_{l=i}^{k-1} p(w_i \to w_k) \\ & \times I(A_{w_k} \to w_{k+1} \dots w_j) I(A_{w_k} \to w_{l+1} \dots w_{k-1}) \\ & \times I(A_{w_i} \to w_{i+1} \dots w_l) \ , \end{aligned} \tag{3b}$$

where the "negative length" string $w_i \dots w_{i-1}$ is understood to represent the null production ϕ. The recursion

[1]The authors would like to thank Joshua Goodman for developing the N^4 procedure, a notable improvement over previous implementations.

is initialized by

$$I(A_{w_i} \rightarrow \phi) = p(w_i \rightarrow \phi) .$$

The above computations involve a double sum and are therefore of order N^2, and there are order N^2 probabilities $I(w_i \rightarrow w_j \ldots w_{i-1})$ and $I(A_{w_i} \rightarrow w_{i+1} \ldots w_j)$, for a total of N^4. (For the Viterbi calculation, one simply selects the largest contribution from the right hand side of equations (3a) and (3b) instead of doing the double sum.)

It is important to note that despite the N^4 behavior, this grammar is in general *faster* than context-free parsing, which is computationally of order N^3. This is because the compute time for context-free parsing also includes a factor proportional to the number of rules in the grammar, which even in simple cases can be in the hundreds. There is no such factor in the computation for this lexicalized grammar—it is effectively replaced by another power of N, which is much smaller.

To see how the probabilities $p(w_1 \rightarrow w_2)$ converge, this model was run through ten iterations of training on approximately 100,000 sentences of ten words or less from the English half of the Canadian Hansard corpus. Some examples of these probabilities follow:

the	*U.*	*tariffs*
.91 ϕ	.52 ϕ	.44 ϕ
	.26 *S.*	.09 *agreement*
	.14 *agreement*	.08 *and*
		.08 *general*
		.08 *on*

As expected, *the* trains strongly to generate the null symbol ϕ. The token *U.* has a strong tendency to generate *S.* for obvious reasons; that it also generates *agreement* is a consequence of the frequent discussion in the corpus of *the U. S. free trade agreement*. This is an example of how the model will find associations between separated words that even a trigram model will not see. The distribution associated with *tariffs* arises from parliamentary debate on the *general agreement on tariffs and trade*.

The simple grammar described above can be considered the starting point for a class of more complex models. One obvious extension is to train the probability distributions for generating to the left and right separately. This corresponds to implementing the grammar

$$A^L_{w_1} \rightarrow A^L_{w_2} w_2 A^R_{w_2} A^L_{w_1} , \quad A^L_{w_1} \rightarrow \phi , \qquad (4a)$$

$$A^R_{w_1} \rightarrow A^R_{w_1} A^L_{w_2} w_2 A^R_{w_2} , \quad A^R_{w_1} \rightarrow \phi . \qquad (4b)$$

Training this grammar on the same text as the original model yields the left probabilities:

the	*U.*	*tariffs*
.96 ϕ	.80 ϕ	.35 ϕ
	.10 *the*	.14 *agreement*
		.12 *general*
		.12 *on*

Again, *the* tends to generate a null. Like most nouns, *U.* has learned to generate a *the* to its left, and the left distribution for *tariffs* includes only those words found typically on its left. The right probabilities for the same words are:

the	*U.*	*tariffs*
.90 ϕ	.36 ϕ	.52 ϕ
	.37 *S.*	.18 *and*
	.19 *agreement*	.17 *trade*
	.07 *free*	

These are also consistent with the results from the original model.

Rearrangement

The next step in LINGSTAT's translation method is a transfer of the parse of each Japanese sentence into a corresponding English parse, giving an English word ordering. This is accomplished through the use of English rewrite rules encoded in the Japanese grammar. Through this encoding, each non-terminal in the Japanese grammar corresponds to a non-terminal in an implied English grammar. The rewrite process just consists of taking the Japanese parse and expanding in this English grammar. As this expansion proceeds, Japanese constructs that are not translated (certain particles, for example) are removed, and tokens for English constructs not represented in the Japanese (such as articles) are introduced.

Annotation/language model

The Japanese words in the reordered sentenced are annotated with (possibly several) candidate English glosses, supplied from an electronic dictionary compiled from various sources. Numbers are translated directly, and katakana tokens (which are usually borrowed foreign words) are transliterated into English. Tokens introduced in the rearrangement step are also glossed; the token indicating an English article is multiply glossed as *the*, *a*, *an*, and *null* (which expands to an empty word).

Inflected Japanese words are glossed by first glossing the root, then applying an English version of the Japanese

inflection to each candidate. This is made difficult by the poor correspondence between Japanese and English inflections: English is inflected for person and number, for example, while in Japanese there are inflections for such constructions as the causative, which require non-local changes in the corresponding English. Japanese inflections also often consist of multiple steps, which means that the English inflections must be compounded. For example, to inflect the verb *to walk* into the past desiderative involves the two step transformation,

to walk → *to want to walk* → *wanted to walk* .

This procedure can produce some unusual results when the number of inflection steps is greater than two.

The final step in the translation process is to apply an English language model to select the best gloss from among the many candidates for each word. In the current system this is done with a trigram model, which makes the choices that maximize the average probability per word. The trigram model used was trained on Wall Street Journal and so has a business bias, partially reflecting the bias of the evaluation texts.

3. RESULTS

The January 1994 ARPA machine translation evaluation has recently been completed. In this test, Dragon used the same translators as in the May 1993 evaluation and provided them with essentially the same interface and online tools. The difference in this evaluation was that the translators were also provided an automatically generated English translation of the Japanese document as a first draft. Manual and machine-assisted translation times were measured, and the automatic output was also submitted for separate evaluation.

Preliminary timing results show a speedup by a factor of 2.4 in machine-assisted vs. manual translation. Because we were using the May 1993 translators, this result may be compared to the May 1993 result; it is essentially unchanged. This suggests that the draft translation was of no significant help to the translators in this evaluation, probably because the quality of automatic output is not high enough to be relied upon.

A quality measurement of the automatic output is not yet available, but we offer one example of a sample translation from the current system. For the following correctly glossed Japanese sentence,

> *(America) (investment bank) NO (Wertheim) (Schroder) WA , (Mitsubishi Trust and Banking Corporation) (to) (same company) NO (stock) NO (4.9%) NO (sell) KOTO WO (decided)*

LINGSTAT produced

> ***waazuhaimu shuroodaa of the America investment bank decided to sell off 4.9% of the shares of the same company to Mitsubishi Trust and Banking Corporation***

Even this simple sentence demonstrates the large amount of rearrangement necessary to render the Japanese into English. This effort is not without errors; a correct translation shows that the word meaning *same company* was mishandled, as was the modifier of *Wertheim Schroder*:

> ***The American investment bank Wertheim Schroder has decided to sell 4.9% of its stock to the Mitsubishi Trust and Banking Corporation***

This sentence is less complex than is typical in a Japanese newspaper article, and therefore LINGSTAT's performance in this case is not representative.

4. FUTURE PLANS

The steps that have the most effect on the quality of the final output translation (at least for Japanese) are the parser and gloss selection modules. The parser in particular is crucial, since it initiates a global rearrangement of the sentence into a sensible English order—a parsing mistake will often render a sentence unintelligible.

The improvements contemplated for the parsing module include more hand work on the coarse context-free grammar to provide more accurate parses, and a general speedup to allow more extensive training. A faster parser would also allow the merging of the two grammars so that they could be trained simultaneously. Attempts to do this have so far resulted in an unacceptable increase in training and parsing time due to the complexity of the algorithm.

The language model used to select glosses in the final translation step must be improved to have more global control. Common mistakes made by the current model include inconsistent glossing of a recurring word and virtually no notion of topic or domain (except on business subjects). Both of these problems are the result of using a language model, trigrams, that uses such restricted context.

The newest version of the system must be ported to Spanish for the next evaluation, scheduled for June. This will require improvements to the Spanish dictionary and de-inflector, an update of the Spanish grammar from the older Spanish system, a lexicalized grammar trained on Spanish text, and Spanish rewrite rules. We intend to use the parallel Spanish-English component of the UN data to provide gloss information.

REFERENCES

[1] J. Yamron, J. Baker, Paul Bamberg, Haakon Chevalier, Taiko Dietzel, John Elder, Frank Kampmann, Mark Mandel, Linda Manganaro, Todd Margolis, and Elizabeth Steele, *LINGSTAT: An interactive, Machine-Aided Translation System,* Proceedings of the ARPA Human Technology Workshop, March 1993.

[2] J.K. Baker, *Trainable Grammars for Speech Recognition,* Speech Communication Papers for the 97th Meeting of the Acoustical Society of America (D.H. Klatt and J.J. Wolf, eds.), pp. 547–550, 1979.

[3] F. Jelinek, J.D. Lafferty, and R.L. Mercer, *Basic Methods of Probabilistic Context-Free Grammars,* in *Speech Recognition and Understanding: Recent Advances, Trends, and Applications,* P. Laface and R. De Mori, eds., Springer-Verlag, Series F: Computer and Systems Sciences, vol. 75, 1992.